1. *Attack Identification, Recognition and Isolation via the Statistical Recognition Unit.*

The statistical recognition unit is responsible for analyzing the attack, identifying its origin(s) and providing operational rules for blocking the attack ~~blockage without~~ disturbing innocent genuine traffic. The basic principle behind the unit's operation is that the pattern of traffic originated at the attack sources during attack time drastically differs from that pattern during normal operation. In contrast, traffic patterns of "innocent" sources during an attack resemble those at normal times. This principle is used to identify the attack sources and ~~recommend on~~provide guidelines for their blockage.

The statistical unit has three major components: a) Classification of the victim traffic into flows b) Learning the traffic patterns of the various victim flows under normal operation conditions, and c) Monitoring the flows traffic pattern at attack time and detecting the attack sources. Below we describe these in detail.

1.1 Network flows and traffic classification

The statistical unit operation is based on classifying the traffic into network flows. A network flow can be viewed as a stream of packets that share the same properties. It is common (e.g., in the Cisco convention) to define a network flow by the following parameters:

i. Source IP address.
ii. Source port.
iii. Destination IP address.
iv. Destination port.
v. Traffic type (TCP / UDP/ SYN).

ZZZ will use either this fine classification or a more coarse classification, ~~according to some of the following~~ guided by the following considerations:

~~a. Elimination of source port.~~

i. **Disregarding source port:** Will be done in the event that source port does not serve as a good separator between malicious and innocent traffic.

~~b.~~ii. **Grouping (aggregating) a set of individual source addresses into one set (e.g., by considering the IP address prefix):** Aggregation, if used, will serve to reduce the number of statistics measured and computed at attack time, thus reducing the processing complexity. Aggregation can be done in a hierarchical manner.

iii. ~~prefix).~~ **Disregarding destination address:** At most cases the unit operates to block attacks oriented on a single target (e.g. www.xyz.com). In these cases, the statistical unit will receive as input ***only*** flows destined to that destination (www.xyz.com). In these cases classifying by destination address is irrelevant.

1.2 Traffic Studying During Peace Time

During "peace time" the unit will actively measure and ~~sniffy~~study the traffic volumes of the various flows. This is done in two major modules:

1. **Traffic volume statistical data collection and classification:** This module operates at "peace" times and is destined to learn the normal traffic volume patterns. This traffic learning is done in either (or both) of the following approaches:
   i. ~~It operates by sampling~~Sampling a **fraction** α of the **packets** ($0 < \alpha \leq 1$) traversing the lines on route to the target and then classifying ~~them~~the packets based on the following parameters:
      א. **Network flow** – classification according to the classification described in the previous sub-section.
   ~~b.~~**Time of the day and day of the ~~week.~~**
      ב. week.

   Note that setting $\alpha = 1$ requires the unit to process every packet and thus imposes high load on it while providing the best statistical measure. Lower values of α reduce the load posed on the unit while potentially somewhat degrading the statistical measure. The fraction α therefore, will be a parameter that will be set so that enough statistical knowledge can be gained without over-loading the system, ~~while system performance does not deteriorate (due to the measurement activity).~~
   This method can be used to classify all traffic type direct to/from the defended targets, and requires sensing ("sniffing") the lines on route to the destination. The sniffing devices must be placed as to measure all traffic, that is, at the network boundary, or at the defendant target proximity.

   ii. Utilizing **server logs** collected by the defended target. These typically contain information about the activity being performed on the target. For example, WEB sites, which are likely to form the main body of potential targets, keep logs that record all the document requests sent to the site (including their source address, time of the day and other parameters). Processing of these logs by ZZZ will yield a very accurate measure of the statistics of network flow volumes (measured in packets per second, as in a) above).

2. The traffic volume data collected will be summarized and will be stored in a database that can be accessed via the various parameters of the flows.

2. **Traffic analysis:** Will be conducted at "peace time" and used to generate statistical summaries of the data collected. In particular the processing will be used to compute **mean** and **variance** of volumes of each of the flows, or aggregates of flows. The analysis may also dynamically change aggregates of flows in order to improve the statistical identification of traffic. The results of this analysis will be stored in a database to be used at attack time. The data base will based on storing flow volumes of large volume sources (or on groups of flows), since most IP addresses generate tiny volumes.

1.3 Traffic Measurement and Analysis at Attack Time

1. Online traffic volume collection at attack time: This module operates during attack times and is responsible to collect the statistics of the traffic at that period. The module receives as input only traffic that is destined to the attacked target(s) and measures its packet rates. Note, that in this sense, its measures are similar to Classification of the traffic the "peace time" measures collected in approach 1a above. The classification of the traffic, in general, is similar to that conducted in "peace time" but may be guided by external mechanisms to achieve better focus. The externalcontrolled/guided by external intervention. Such intervention will be enacted if some additional knowledge on the attack type is gained from other sources (e.g., mechanisms can originate from decisions made by ZZZ or from administrative information entered to the system (e.g., the identity, namely destination address, of the attacked target), human-aided identification) and can be utilized by the unit.

2. Attack Analysis: Will be conducted during attack time and will be responsible to compare the historically collected statistical data with the current traffic volume and generate rules for traffic blockage. The output of this unit, in general, will consist of a list of items for each of which three parameters will be provided:
a. **Network flow**, identified by a combination of source IP address (can be prefixed), destination IP address, destination port number, protocol type.
b. **Duration,** identifying the duration for which that class will be blocked.

The analysis will be based on the statistical parameters of the data and will aim at keeping the attacked destination at normal loads by blocking the most "suspected" traffic streams. Blocking rules will be based on maximizing the likelihood of blocking malicious data while minimizing the likelihood of blocking innocent data.

1.4 Statistical Recognition of Data "Innocence"

ZZZ will use two major properties of network flows to identify whether they are malicious or innocent. These are: a) Traffic pattern, and b) Traffic volume. Below we describe the recognition approaches based on these factors.

1.4.1 Recognition of Traffic Pattern

~~Several aspects of traffic pattern will be examined:~~

~~1) Source "IP geography" proximity: Sources will be classified into classes that resemble the "IP geography", that is IP addresses that reside on similar networks (similar IP address prefix) will be classified in the same class. A class that will generate a relatively large volume of requests will be suspected as being malicious. Note that such "malicious classes" are likely to form if the attacker planted a collection of daemons in the same network.~~

~~2) Periodicity: Sources will be examined for the periodicity of their requests. It is likely that malicious sources (unless being very sophisticated) will act in a relatively periodic manner, while innocent sources act in more random fashion.~~

~~3) Packet Properties: Sources will be examined for repetitive properties of their packets. For example the distribution of packet size. It is likely that malicious sources (unless very sophisticated) will generate packets of identical properties (e.g. – all packets of same size) while innocent sources will generate packets of more random nature.~~

Several aspects of traffic pattern will be examined:

1) **Source "IP geography" proximity**: Sources will be classified into classes that resemble the "IP geography", that is IP addresses that reside on similar networks (similar IP address prefix) will be classified in the same class. A class that will generate a relatively-large volume of requests will be suspected as being malicious. Note that such "malicious classes" are likely to form if the attacker planted a collection of daemons in the same network.
2) **Periodicity:** Sources will be examined for the periodicity of their requests. It is likely that malicious sources (unless being very sophisticated) will act in a relatively periodic manner, while innocent sources act in more random fashion.
3) **Packet Properties:** Sources will be examined for repetitive properties of their packets. For example the distribution of packet size. It is likely that malicious sources (unless very sophisticated) will generate packets of identical properties (e.g. – all packets of same size) while innocent sources will generate packets of more random nature.

1.4.2 Recognition of Traffic Volume

Traffic volume recognition will be used to identify malicious sources that transmit **large volumes** of data which **significantly differ** from their normal volume. Specifically, we classify Internet data sources to *small sources* and *large sources*. The former relates to individual IP addresses whose traffic volume is normally tiny. The latter relates to Proxy traffic or Spider traffic[1] whose volume is drastically higher.

[1] The traffic volume resulting from a Spider access is normally higher than that of a single human user, especially on large WEB sites. The reason is that a spider scans the whole site, leading to hundred or thousands of requests while a human client requests tens of pages or less, on average.

ZZZ will keep individual volume history for each of the large sources. Individual history will not be kept for the small sources; rather a single fixed small number (related to their mean volume averaged over all these sources) will be recorded. At attack time the traffic volumes of individual flows will be measured and compared to their recorded volume. Flows whose volume will drastically differ (upwards) from their recorded measure will be marked as being malicious.

The mathematical formulation of this procedure is as follows: Given are K classes of flow of traffic, indexed *1, 2, ..., K,* and characterized by the mean (μ_i) and the variance (σ_i) of their historical volume, and by their current volume (X_i). We would like to identify the classes which mostly deviate from their expected volume. Let $Y_i = (X_i - \mu_i)/\sigma_i$. We will sort the classes by the value of Y_i and will recommend to block (eliminate) the classes with the largest values of Y_i.

1.4.2.1 Time accumulating traffic volume recognition and "controlled" denial of service

It is important to recognize that the effectiveness of volume recognition increases with the time duration along which it is implemented. This is correct since the variance of total data volume generated by a source during a period of duration T decreases in T. For example, it is expected that the average amount of traffic generated by a small source during a period of 1 hour will be *very small.* However, at certain epochs, it is expected that the average amount of traffic generated by the same source during a period of 1 minute can be rather large. (up to 60 times larger than that of the 1 hour average).

For this reason ZZZ will implement the following unique recognition and traffic screening mechanism. For source i, let $S_i(t)$ denote the amount traffic generated by the source during the interval *(0,t)* (where we assume that the attack starts at time 0). We then set at time t: $X_i(t) = S_i(t)/t$ and apply the above screening mechanism.

This mechanism has the following properties:

1. For a small value of t (that is, at the attack beginning moments) a sophisticated attacker might cause significant number of innocent users denial of service. This is due to the fact that the attacker may inflict a load that resembles that of an innocent client, and thus the attacker is not distinguishable from the innocent client. At this stage, ZZZ will may block some innocent clients and some attackers. Using this action, for a short period some innocent clients may be denied of service but ZZZ protects the site from going down!
2. As t increases the denial of service conducted by ZZZ will be acted more and more on the

malicious sources and less and less on the innocent sources. This is since the malicious sources have posed large amount of accumulated load. Thus as time progresses less and less innocent clients are denied of service. In fact, after relatively short period all malicious sources will be denied of service while the innocent sources will receive full regular service.

Example: Consider the traffic volume generated on the web site of the Nagano server (Feb 98). It had 11,665,713 requests made over a period of 24 hours by 59,582 clients. Assuming uniform distribution of clients over the day, this implies about 2500 clients per hour and 500 clients per 12-minute interval. An attacker who uses 500 sophisticated daemons (which imitate a normal client) will look innocent at the first 12 minutes interval. At this period ZZZ will block 50% of the innocent clients and 50% of the daemons. However, after 24 minutes the daemons will generate significantly more traffic than an innocent client and thus almost all of the traffic blocked will be that of daemons.